\*:prefix = "( :let | :expression ) ( ';' | ',')? ( :let | :as | :keyExp ) \* "

prefix -Directive selector | Attribute name
let - let localVariable ( = exportKey)? [;]?
expression - Input Property binding exp
as - exportKey as localVariable [;]?
keyExp - inputProp [:]? expression [;]?

# Either start with a **let** or **expression**

let - Context & Template Input Variable

\*player="let item"

Context **\$implicit** property (Default exported key)

\*player="let role=type"

Context property (Exported key)

# **Angular Expression**

Input property binding (Same name as Directive selector)

\*player="name" Expression: Direct Prop binding

\*player="name.slice(0, 4)" Expression: Prop binding

\*player="name.slice(0, 4) | uppercase" Expression



We cannot set the **Input property** other than the same name as the **Directive selector** 

\*:prefix = "( :let | :expression ) ( ';' | ',')? ( :let | :as | :keyExp ) \* "

Start with a **let** or **as** or **keyExp** & this whole expression can happen **multiple** times

After each **let**, **as** or **keyExp** we can have optional **comma** or **semi-colon** 

type as role

let - let localVariable ( = exportKey)? [;]?
as - exportKey as localVariable [;]?
keyExp - inputProp [:]? expression [;]?

let item; let item, let item

let role=type; let role=type

let item, let role=type

let item; let role=type

as - Context & Template Input Variable

type as role, points as score

type as role;

as can only be used for defined keys not for \$implicit (default) key

type as role,

let item, type as role, points as score

let item, type as role, let score=points

**keyExp** - Input Property Binding
Selector + PascalCase (Property Name)

xCoord: 12; xCoord: 12, xCoord: 12

xCoord: 12, yCoord: 8 xCoord: 12; yCoord: 8

xCoord 12 yCoord 8

Combined

let item, type as role, let score=points; xCoord: 12, yCoord: 8

type as role, xCoord: 12, yCoord: 8, let item, let score=points

let item, type as role; xCoord 12, yCoord 8;

3

\*:prefix = "(:let | :expression)(';'|',')?(:let | :as | :keyExp) \* "

2

# \*player="let item; let role=type"

Context **\$implicit** property (Default exported key)

Context property (Exported key)

### \*player="name; let role=type"

Input property binding (Same name as Directive selector)
Context property (Exported key)

# \*player="name; let item, type as role"

Input property binding (Same name as Directive selector)

Context **\$implicit** property (Default exported key)

Context property (Exported key)

# \*player="name; let item, let role=type; xCoord: 12, yCoord: 8"

Input property binding (Same name as Directive selector)

Context **\$implicit** property (Default exported key)

Context property (Exported key)

Other Input bindings

prefix -Directive selector | Attribute name
let - let localVariable ( = exportKey)? [;]?
expression - Input Property binding exp
as - exportKey as localVariable [;]?
keyExp - inputProp [:]? expression [;]?

### \*player="let item xCoord 12 yCoord 8; let role=type;

Context **\$implicit** property (Default exported key)

Other Input bindings

Context property (Exported key)

### **NOTE**

1st segment must be an Input property with the same name as Directive selector or a Context property that is a Template Input Variable.

\*:prefix = "( :let | :expression ) ( ';' | ',')? ( :let | :as | :keyExp ) \* "

# \*player="name as n"

Input property binding (Same name as Directive selector) - Exported in Context

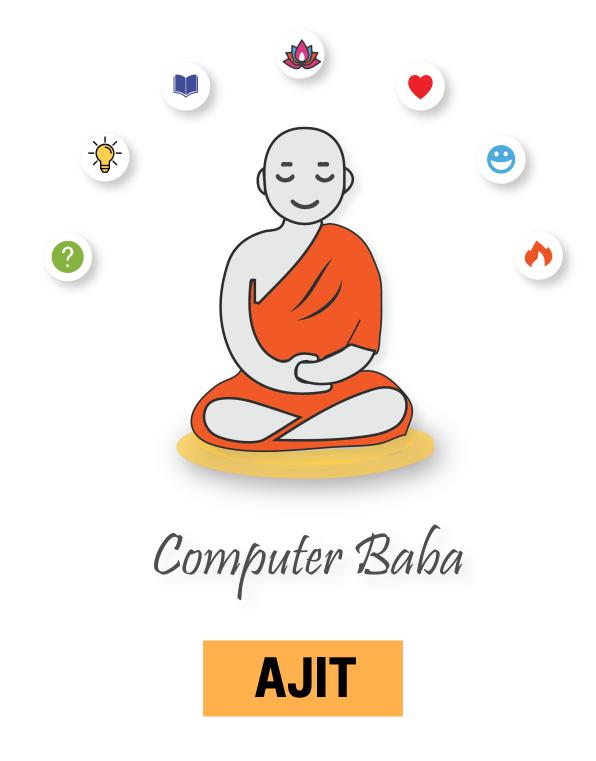
\*player="name as n; xCoord 12 as x"

Input property binding (Same name as Directive selector) - Exported in Context
Input property binding - Exported in Context

prefix -Directive selector | Attribute name
let - let localVariable (= exportKey)? [;]?
expression - Input Property binding exp
as - exportKey as localVariable [;]?
keyExp - inputProp [:]? expression (as localVariable)? [;]?

### NOTE

Input properties must be exported in Context to use them as Template Input Variable.



Youtube Channel https://www.youtube.com/c/ComputerBabaOfficial

Twitter https://twitter.com/akacomputerbaba

Discord Server https://discord.gg/9V4VTDM